

# SingleParticleGun: Shoots single HepMC particles into Athena transient store

Marjorie Shapiro (MDShapiro@lbl.gov), I Hinchliffe (I\_Hinchliffe@lbl.gov)

March 7, 2003

## Updated for Generators in release 6.0.0, March 2003.

This package runs from within Athena, puts single particles into the transient store in HepMC format. See the file GeneratorModules.ps for general information. The note refers only to SingleParticleGun specific material.

Particles are specified by their PDG ID code and either by ( $P_T$ ,  $\eta$  and  $\phi$ ) or by ( $E$ ,  $\theta$  and  $\phi$ ).

The module is activated from the jobOptions service.

See the example in **GeneratorModules/test/share/jobOptions.single.txt**. The specific lines are

```
ApplicationMgr.DLLs += { "DumpMC", "SingleParticleGun" };
ApplicationMgr.TopAlg = {"SingleParticleGun", "DumpMC"};
ApplicationMgr.EvtMax = 5;
```

Number of events to be processed (default is 10)

The vertex can be specified as follows

```
SingleParticleGun.VertexX=5.;
SingleParticleGun.VertexY=5.;
SingleParticleGun.VertexZ=5.;
SingleParticleGun.Time=5.;
```

The ( $P_T$ ,  $\eta$  and  $\phi$ ) mode (**PtMode**) is specified by

```
SingleParticleGun.Mode=2;
```

In this case, the  $P_T$ ,  $\eta$  and  $\phi$  ranges are set from the jobOptions as follows

```
SingleParticleGun.MinPt=5.;
SingleParticleGun.MinPhi= 0.;
SingleParticleGun.MinEta= -5.;
SingleParticleGun.MaxPt= 100.;
SingleParticleGun.MaxPhi= 6.283185;
SingleParticleGun.MaxEta= 3.;
SingleParticleGun.PdgCode= 211;
```

The particle type is set using the PDG code. The particles are then generated with the with either flat or Gaussian distribution set by

```

SingleParticleGun.ModePt= {i};
SingleParticleGun.ModeEta= {i};
SingleParticleGun.ModePhi= {i};

```

and  $i$  is 2 for Gaussian and 3 for uniform distribution A “SingleParticleGunGenMode::FixedMode” is also supported by selecting 1 for  $i$ . In this case, the particle is at fixed values of  $P_T$   $\eta$  and  $\phi$  set by

```

SingleParticleGun.Pt= 5.
SingleParticleGun.Phi= 0
SingleParticleGun.Eta= 5.

```

The defaults are FixedMode for ModePt and uniform for ModeEta and ModePhi. In the case of Gaussian mode the widths of the Gaussians should be specified by

```

SingleParticleGun.SigmaPt= 5. ;
SingleParticleGun.SigmaPhi= 2. ;
SingleParticleGun.SigmaEta= 0.6;

```

The  $(E, \theta$  and  $\phi)$  mode (**EMode**) is specified by

```
SingleParticleGun.Mode=1;
```

In this case, the  $E, \theta$  and  $\phi$  ranges are set from the jobOptions as follows

```

SingleParticleGun.MinE=5. ;
SingleParticleGun.MinPhi= 0. ;
SingleParticleGun.MinTheta= 0. ;
SingleParticleGun.MaxE= 100. ;
SingleParticleGun.MaxPhi= 6.283185;
SingleParticleGun.MaxTheta= 1.1;
SingleParticleGun.PdgCode= 211;

```

The particle type is set using the PDG code The particles are then generated with the with either flat or Gaussian distribution set by

```

SingleParticleGun.ModeE= {i};
SingleParticleGun.ModeTheta= {i};
SingleParticleGun.ModePhi= {i};

```

and  $i$  is 2 for Gaussian and 3 for uniform distribution A “SingleParticleGunGenMode::FixedMode” is also supported by selecting 1 for  $i$ . In this case, the particle is at fixed values of  $E \theta$  and  $\phi$  set by

```

SingleParticleGun.E= 5.
SingleParticleGun.Phi= 0.
SingleParticleGun.Theta= 1.6

```

In the case of Gaussian mode the widths of the Gaussians should be specified by

```

SingleParticleGun.SigmaE= 5. ;
SingleParticleGun.SigmaPhi= 2. ;
SingleParticleGun.SigmaTheta= 0.6;

```

The defaults are FixedMode for ModeE and ModeTheta and FlatMode for ModePhi.  
The ( $E$ ,  $\eta$  and  $\phi$ ) mode (**EtaMode**) is specified by

```
SingleParticleGun.Mode=3;
```

In this case, the  $E$ ,  $\eta$  and  $\phi$  ranges are set from the jobOptions as follows

```
SingleParticleGun.MinE=5.;  
SingleParticleGun.MinPhi= 0.;  
SingleParticleGun.MinEta= -3.;  
SingleParticleGun.MaxE= 100.;  
SingleParticleGun.MaxPhi= 6.283185;  
SingleParticleGun.MaxEta= 3.;  
SingleParticleGun.PdgCode= 211;
```

The particle type is set using the PDG code The particles are then generated with the with either flat or Gaussian distribution set by

```
SingleParticleGun.ModeE= {i};  
SingleParticleGun.ModeEta= {i};  
SingleParticleGun.ModePhi= {i};
```

and  $i$  is 2 for Gaussian and 3 for uniform distribution A “SingleParticleGunGenMode::FixedMode” is also supported by selecting 1 for  $i$ . In this case, the particle is at fixed values of  $E$   $\eta$  and  $\phi$  set by

```
SingleParticleGun.E= 5.  
SingleParticleGun.Phi= 0.  
SingleParticleGun.Eta= 1.6
```

In the case of Gaussian mode the widths of the Gaussians should be specified by

```
SingleParticleGun.SigmaE= 5.;  
SingleParticleGun.SigmaPhi= 2.;  
SingleParticleGun.SigmaEta= 0.6;
```

The defaults are FixedMode for ModeE and FlatMode for ModePhi and ModeEta.

The overall defaults are the following which make  $pi^+$  with  $P_T$  of 5 GeV, uniform in  $\pi$  and in the range  $-4 < \eta < 4$  with a vertex at (0,0,0,0).

```
declareProperty("Mode",m_Emode = SingleEnergyMode::PtMode);  
declareProperty("Pt",m_requestedPt = 5.0);  
declareProperty("E",m_requestedE = 5.0);  
declareProperty("Phi",m_requestedPhi = 0.0);  
declareProperty("Eta",m_requestedEta = 0.0);  
declareProperty("Theta",m_requestedTheta = 0.0);  
declareProperty("Vertex-X",m_requestedX = 0.0);  
declareProperty("Vertex-Y",m_requestedY = 0.0);  
declareProperty("Vertex-Z",m_requestedZ = 0.0);  
declareProperty("Time",m_requestedT = 0.0);  
declareProperty("MinPt",m_minPt = 1.);  
declareProperty("MinEta",m_minEta = -4.0);
```

```

declareProperty("MinEta",m_minTheta = 0.);
declareProperty("MinPhi",m_minPhi = 0.);
declareProperty("MaxPt",m_maxPt = 100.);
declareProperty("MaxEta",m_maxEta= 4.0);
declareProperty("MaxTheta",m_maxTheta=pi);
declareProperty("MaxPhi",m_maxPhi = twopi);
declareProperty("SigmaPt",m_sigmaPt = 0.1);
declareProperty("SigmaE",m_sigmaE = 0.1);
declareProperty("SigmaEta",m_sigmaEta= 0.1);
declareProperty("SigmaTheta",m_sigmaTheta= 0.1);
declareProperty("SigmaPhi",m_sigmaPhi = 0.1);
declareProperty("ModePt",m_PtGenMode = SingleParticleGunGenMode::FixedMode);
declareProperty("ModeE",m_EGenMode = SingleParticleGunGenMode::FixedMode);
declareProperty("ModeTheta",m_ThetaGenMode=SingleParticleGunGenMode::FixedMode);
declareProperty("ModeEta",m_EtaGenMode=SingleParticleGunGenMode::FlatMode);
declareProperty("ModePhi",m_PhiGenMode=SingleParticleGunGenMode::FlatMode);
declareProperty("PdgCode",m_pdgCode=211);

```

**Units:** All energy and momenta are in GeV, distances and time ( $c\tau$ ) are in mm.

### Random Numbers

SingleParticleGun is using the AtRndmGenSvc Athena Service for the necessary random numbers. This service is using the RanecuEngine of CLHEP, and is based on the “stream” logic, each stream being able to provide an independent sequence of random numbers. SingleparticleGun is using one stream, SINGLE, for the event generation. The user can set the initial seeds of the stream via the following option in the jobOption file.

```
AtRndmGenSvc.Seeds = {“SINGLE 2345533 9922199”};
```

The above sets the seeds of the SINGLE stream to 2345533 and 9922199. If the user will not set the seeds of a stream then the AtRndmGenSvc will use default values.

The seeds of the Random number service are saved for each event in the HepMC Event record and they are printed on screen by DumpMC. In this way an event can be reproduced easily. The user has to rerun the job by simply setting the seeds of the SINGLE stream to the seeds of that event.

Additionaly the AtRndmGenSvc is dumping into a file (AtRndmGenSvc.out) the seeds of all the streams at the end of the job. This file can be read back by the service if the user set the option

```
AtRndmGenSvc.ReadFromFile = true;
```

(default = false). In this case the file AtRndmGenSvc.out is read and the streams saved in this file are created with seeds as in this file. The name of the file to be read can be set by the user via the option

```
AtRndmGenSvc.FileToRead = MyFileName;
```

The above file is also written out when a job crashes. **This last option (when job crashing) is currently not working, waiting for a modification in Athena.**

The `GeneratorModules/share/jobOptions.single.txt` contains the above options.